Comparing Variable Width Backtracking and Metaheuristics, Experiments with the Maximum Diversity Problem

[Extended Abstract]

José-Matías Cutillas-Lozano, Miguel-Ángel Franco, Domingo Giménez
Departamento de Informática y Sistemas, University of Murcia, Spain
{josematias.cutillas,mafg1,domingo}@um.es

ABSTRACT

The solution of large NP-hard optimization problems is traditionally approached with heuristic methods, which can provide satisfactory solutions with low execution times, but without guaranteeing the eventual evaluation of all the possible solutions. Some exact methods, like backtracking, analyse all the solutions, so obtaining the optimum solution, but at the expense of unaffordable execution times. This paper compares the application to the Maximum Diversity Problem of metaheuristic methods and a modification of Backtracking, which begins by analysing a part of the solutions tree and in successive steps increases the search so that the whole solutions space is, eventually, explored.

CCS Concepts

•**Theory of computation** → **Backtracking;** *Theory of randomized search heuristics;* •**Computing methodologies** → Shared memory algorithms;

Keywords

Optimization problems, NP-hard problems, Metaheuristics, Backtracking, Maximum diversity problem

1. INTRODUCTION

NP-hard problems can be solved with exact methods only for very small instances, and so they are traditionally approached through heuristic and metaheuristic methods [1, 2], which are tuned for the problem in hand so that satisfactory solutions can be obtained in not very large execution times. On the other hand, some exact methods, like backtracking or branch and bound, analyse all the possible solutions, so obtaining the optimum one, but at the expense of an infeasible execution time. A modified backtracking is applied to the Maximum Diversity Problem (MDP) [3]. We call the scheme Variable Width Backtracking (VWB), and it uses a time-dependent distribution of the search of the different branches in the tree. The results obtained with VWB are compared with those with Genetic Algorithms (GA) and Scatter Search (SS).

GECCO '15 Companion, July 11-15, 2015, Madrid, Spain

ACM ISBN 978-1-4503-3488-4/15/07
DOI: http://dx.doi.org/10.1145/2739482.2764883

2. VWB FOR MDP

The MDP consists of selecting a subset of m elements from a set of n elements so that the sum of the distances between the chosen elements is maximized:

$$\textbf{Maximize} \quad \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}x_ix_j \tag{1}$$

subject to

$$\sum_{i=1}^{n} x_i = m, \qquad x_i = \{0,1\}, \qquad 1 \le i \le n \tag{2}$$

The key idea of VWB is to obtain for each node in the tree lower and upper bounds (LB and UB) of the value of the optimum solutions reachable, and to divide the work in successive steps, s, with the work in each step consisting of a backtracking with the lower and upper bounds of each node in this step (LB_i and UB_i for step i) obtained in the interval of the complete bounds, with the interval in a step including the interval of the previous step, and with the interval of the last step equal to the complete interval: $[LB_i, UB_i] \subset [LB_{i+1}, UB_{i+1}]$ and $[LB_s, UB_s] = [LB, UB]$. So, the tree is explored in successive steps, with possible improvement of the best solution in each step, and the optimum solution of the last step is that of the problem. Furthermore, the search in each step is scattered throughout the tree, in an attempt to avoid the temporal concentration of the search in certain areas of the tree, which could delay finding satisfactory solutions. Unfortunately, the search in step $i+1$ must include the nodes generated in the previous steps, and so the huge execution time of backtracking methods is incremented, and the only possible advantage of VWB is the early generation of satisfactory results. The scheme is shown in Algorithm 1.

3. COMPARISON OF VWB AND METAHEURISTICS FOR THE MDP

The behaviour of VWB is compared with that of GA and SS. For small problems, for which the optimum solution can

Algorithm 1 Scheme of VWB

Ensure: The optimum solution and its value in POS and POV
$POS = Initial\ solution$
$POV = Initial\ value$
for $i = 1, \ldots, s$ **do**
  PartialBacktracking(i)
end for

Table 1: Execution time (in seconds) to achieve the optimum solution, for two small instances of MDP, with metaheuristics (GA and SS) and several backtracking implementations.

method:	GA	SS	Back1	Back2
$n = 30$, $m = 12$	0.018	0.095	13.97	44.14
$n = 30$, $m = 18$	0.056	0.222	6.62	16.95
method:	VWB10	VWB20	VWB40	VWB80
$n = 30$, $m = 12$	1.71	2.22	3.68	4.55
$n = 30$, $m = 18$	5.23	2.86	5.00	9.35

be obtained with backtracking in a low execution time, Table 1 shows the execution time of various methods to achieve the optimum solution. $n = 30$, the number of elements to be selected is $m = 12$ and $m = 18$, and the total number of combinations is $8,649,225$ in both cases, which means an affordable number of nodes for backtracking. The metaheuristic methods obtain the optimum solution with few iterations (2 or 3 GA and 1 or 2 SS) and with low execution times. Two backtracking implementations are considered; `Back1` does not associate upper bounds to the nodes, so all the possible solutions are explored; `Back2` associates bounds to the nodes. The times of the backtracking implementations are higher than those of the metaheuristics, but we are sure the optimum solution is found. The columns `VWB`s correspond to implementations of VWB with `Back2` as the basic backtracking and with 10, 20, 40 and 80 steps. VWB gives lower execution times than the normal backtracking, with the best results for a moderate value for s (10 or 20). VWB does not improve the times of metaheuristics, but it can be useful for the early finding of satisfactory solutions. Figure 1 compares the fitness obtained at different times with GA, `Back1`, `Back2` and VWB with $s = 10$. The problem size is $n = 150$, $m = 45$, with a total of $4.4167 \cdot 10^{38}$ possible configurations, which is an unaffordable number of nodes for backtracking. GA outperforms VWB10 after 25 seconds of execution, which confirms our idea of the use of VWB for early finding of good solutions.

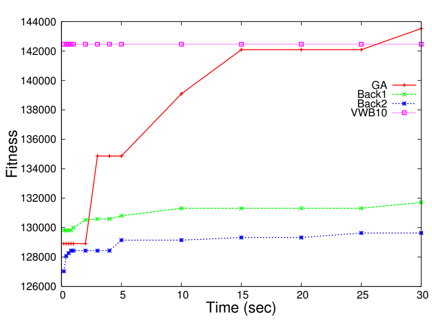

Figure 1: Fitness obtained at different times with VWB10, GA, `Back1` and `Back2`.

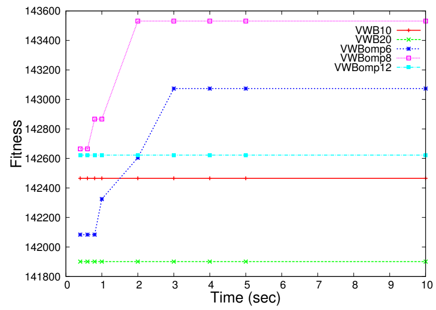

Figure 2: Fitness obtained at different times with VWB10 and VWB20 and with the OpenMP implementation of VWB for 6, 8 and 12 threads.

A multicore version of VWB is aimed at finding early solutions with high fitness. It uses OpenMP to start several threads, with each thread executing VWB with different values of s. Figure 2 compares the fitness obtained with VWB10 and VWB20 with those of the multiple executions of VWB with the OpenMP implementation, for 6, 8 and 12 threads. The OpenMP implementations gives better results than the sequential VWB, with the best values obtained with 8 threads. The shared-memory implementation of VWB gives satisfactory results without having to determine a satisfactory value for s in VWBs.

4. CONCLUSIONS

A Variable Width Backtracking has been compared with metaheuristics for the solution of the Maximum Diversity Problem. Metaheuristics are preferable to exact methods for NP-hard problems, but VWB can be used to obtain satisfactory results earlier than metaheuristics. To do so, it is necessary to select the number of VWB steps. Using the parallel capacity of todays computational nodes, an OpenMP version of VWB with multiple executions of VWB with different numbers of steps can be used to run VWB without selecting the best value for the number of steps of VWB.

Preliminary but promising results are reported, but more exhaustive experimentation is necessary, with MDP and other optimization problems. Furthermore, VWB could be integrated with metaheuristic methods to generate initial satisfactory solutions in the initialization phase of metaheuristics.

5. ACKNOWLEDGMENTS

Supported by Spanish MINECO and European Commission FEDER funds, TIN2012-38341-C04-03.

6. REFERENCES

[1] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli. Hybrid metaheuristics in combinatorial optimization: A survey. *Appl. Soft Comput.*, 11(6):4135–4151, 2011.

[2] F. Glover and G. A. Kochenberger. *Handbook of Metaheuristics*. Kluwer, 2003.

[3] R. Martí, A. D. M. Gallego, and E. G. Pardo. Heuristics and metaheuristics for the maximum diversity problem. *J. Heuristics*, 19(4):591 – 615, 2013.